What is claimed is:

1. A method of performing an arbitrary permutation in a programmable processor comprising the steps of:

a. defining bit positions in a source sequence of bits to be permuted in a source register for a group of bits in a destination register;

b. determining a permutation instruction with said bit positions to assemble bits from said source sequence of bits;

c. performing said permutation instruction for inserting said assembled bits into a destination register as determined by said bit positions; and

d. repeating steps a. through c. for groups of bits in said destination register,

wherein after a final permutation instruction a desired permutation of said source register is determined and said determined permutation instructions form a permutation instruction sequence.

2. The method of claim 1 wherein step d repeats steps a. through c. for all non-overlapping said groups of bits in said destination register.

3 The method of claim 1 wherein said permutation instruction comprises a first parameter indicating which *k* bits in said destination register will change, a reference to said source register which contains said source sequence of bits to be permuted, a reference to a configuration register which contains configuration bits for indicating which said bits in said source register are assembled and a reference to said destination register.

4. The method of claim 3 wherein in said destination register said *k* bits specified by said first parameter are updated and all other bits in said destination register are set to zero.

5. The method of claim 1 wherein each of said *k* bits in said final permutation is determined by lg*n* bits to specify which bit in said source register to change.

6. The method of claim 3 wherein in said destination register said *k* bits specified by said first parameter are updated.

7. The method of claim 1 wherein said programmable processor is a microprocessor, digital signal processor, media processor, multimedia processor, cryptographic processor, network processor, or programmable System-on-a-Chip(SOC).

8. A method of performing an arbitrary permutation of a source sequence of bits into a final arrangement of bits in a programmable processor comprising the steps of:

a. determining the final arrangement of bits of an arbitrary permutation;

b. defining an intermediate sequence of bits that said final arrangement of bits is transformed from;

c. determining a permutation instruction for transforming said intermediate sequence of bits into said final arrangement of bits by dividing said –intermediate sequence into a first group and a second group and combining said first group and said second group; and

d. repeating steps b. and c. using said determined intermediate sequence of bits from step b. as said final arrangement of bits in step c. until an intermediate sequence of bits is obtained that is the same as the source sequence of bits,

wherein the determined permutation instructions, in reversed order, form a permutation instruction sequence.

9. The method of claim 8 wherein said permutation instruction comprises a reference to a source register which contains said source sequence of bits or said

intermediate sequence of bits, a reference to a configuration register which contains control bits, and a reference to a destination register to which said final arrangement of bits or said desired arbitrary permutation is placed.

10. The method of claim 9 wherein bits in said arrangement are divided into said first group if said control bit is 0 and into said second group if said control bit is 1.

11. The method of claim 10 wherein said first group and said second group are combined by putting said first group to the left of said second group.

12. The method of claim 8 wherein at most log *n* said permutation instructions are included in said permutation instruction sequence wherein *n* is the number of subwords in said sequence of bits, each said subword comprising one or more bits.

13. The method of claim 8 wherein said programmable processor is a microprocessor, digital signal processor, media processor, multimedia processor, cryptographic processor, network processor, or programmable System-on-a-Chip(SOC).

14. A method of performing an arbitrary permutation at a source sequence of bits, called the initial arrangement, in a programmable processor comprising the steps of:

a. determining the final arrangement of a sequence of bits to be permuted;

b. determining a number of monotonically increasing sequences (MIS) in said arrangement;

c. determining a first group of MISes and a second group of MISes;

d. combining each element of said first group sequentially with each element of said second group to form a merged group;

e. sorting said merged group in increasing numerical order for determining an intermediate arrangement from said sorted merged group;

f. determining control bits for said intermediate permutation instruction;

if said intermediate arrangement is a single monotonically increasing sequence said intermediate arrangement is the initial arrangement, wherein said determined intermediate permutation instructions form a permutation instruction sequence; and

if said intermediate arrangement is not a single monotonically increasing sequence repeating steps b through f.

15. The method of claim 14 wherein at most lg*n* said permutation instructions are included in said permutation instruction sequence, wherein *n* is the number of subwords in said sequence of bits, each said subword comprising one or more bits.

16. The method of claim 14 wherein said programmable processor is a microprocessor, digital signal processor, media processor, multimedia processor, cryptographic processor, network processor, or programmable System-on-a-Chip(SOC).

17. A method of performing an arbitrary permutation of a source sequence of bits in a programmable processor said source sequence of bits is packed into a plurality of source registers comprising the steps of:

a. dividing bits of a first of said source registers to be placed in a first destination register into a first group and bits of said first of said source registers to be placed in a second destination register into a second group with a first GRP permutation instruction;

b. dividing bits of a second of said source registers to be placed in said first destination register into a first group and bits of said second of said source registers to be placed in a second destination register into a second group with a second GRP permutation instruction;

c. placing bits of said first group of said first of said source registers and said bits of said first group of said second of said source registers into said first destination register;

d. placing bits of said second group of said first of said source registers and said second group of said second of said registers into said second destination register;

e. defining a sequence of bits of said first destination register as a first source sequence of bits and a sequence of bits of said second destination register as a second source sequence of bits;

f. defining an intermediate sequence of bits that each of said first source sequence of bits and said second source sequence of bits is transformed;

g. determining a GRP permutation instruction for transforming said first source sequence of bits and said second source sequence of bits into respective said intermediate sequence of bits; and

h. repeating steps f. and g. using said determined intermediate sequence of bits from step g. as said source sequence of bits in step f. until a respective desired sequence of bits is obtained for said first source sequence of bits and said second source sequence of bits,

wherein the determined permutation instructions form a permutation instruction sequence.

18. The method of claim 17 wherein said programmable processor is a microprocessor, digital signal processor, media processor, multimedia processor, cryptographic processor, network processor, or programmable System-on-a-Chip(SOC).

19. The method of claim 17 wherein at most $2\lg n + 4$ instructions are included in said permutation instruction sequence, wherein n is the number of subwords in said sequence of bits, each said subwords comprising one or more bits.

20. A system of performing an arbitrary permutation in a programmable processor comprising:

means for defining bit positions in a source sequence of bits to be permuted in a source register for a group of bits in a destination register;

means for determining permutation instructions with said bit positions to assemble bits from said source sequence of bits into one or more intermediate sequencesof bits until a desired sequence is obtained;

means for performing said determined permutation instructions for inserting said assembled bits into a destination register as determined by said bit positions for each of said one or more intermediate sequences of bits or said desired sequence;

wherein after a final permutation instruction a desired permutation of said source register is determined and said determined permutation instructions form a permutation instruction sequence.

21. The system of claim 20 wherein said permutation instruction comprises a first parameter indicating which k bits in said destination register will change, a reference to said source register which contains said source sequence of bits to be permuted, a reference to a configuration register which contains configuration bits for indicating which said bits in said source register are assembled and a reference to said destination register.

22. The system of claim 21 wherein in said destination register said k bits specified by said first parameter are updated and all other bits in said destination register are set to zero.

23. The system of claim 20 wherein each of said k bits in said final permutation is determined by lg$n$ bits to specify which bit in said source register to change.

24. The system of claim 21 wherein in said destination register said k bits permuted by said first parameter are updated.

25. The system of claim 20 wherein said programmable processor is a microprocessor, digital signal processor, media processor, multimedia processor, cryptographic processor, network processor, or programmable System-on-a-Chip(SOC).

26. A system of performing an arbitrary permutation at a source sequence of bits in a programmable processor comprising the steps of:

means for determining an initial and final arrangement of a source sequence of bits;

means for defining one or more intermediate sequence of bits that said initial arrangement of said source sequence of bits is transformed into until a desired sequence is obtained;

means for determining permutation instructions for transforming said source sequence of bits into for each of said one or more intermediate sequence of bits or said desired sequence by dividing said arrangement into a first group and a second group and combining said first group and said second group;

wherein the determined permutation instructions form a permutation instruction sequence.

27. The system of claim 26 wherein said permutation instruction comprises a reference to a source register which contains said arrangement, a reference to a configuration register which contains configuration bits and a reference to a destination register to which the intermediate sequence of bits or said desired sequence of bits is placed.

28. The system of claim 27 wherein bits in said arrangement are divided into said first group if said configuration bit is 0 and into said second group if said configuration bit is 1.

29. The system of claim 28 wherein said first group and said second group are combined by putting said first group to the left of said second group.

30. The system of claim 28 wherein at most lg$n$ said permutation instructions are included in said permutation instruction sequence wherein n is the number of subwords in said sequence of bits, each said subword comprising one or more bits.

31. The system of claim 26 wherein said programmable processor is a microprocessor, digital signal processor, media processor, multimedia processor, cryptographic processor, network processor, or programmable System-on-a-Chip(SOC).

32. A system of performing an arbitrary permutation at a source sequence of bits in a programmable processor comprising:

means for determining an initial and final arrangement of a sequence of bits to be permuted;

means for determining a number of monotonically increasing sequences (MIS) in said arrangement;

means for determining a first group of MISes and a second group of MISes;

means for combining each element of said first group sequentially with each element of said second group to form a merged group;

means for sorting said merged group in increasing numerical order for determining an intermediate arrangement from said sorted merged group;

means for determining control bits for said intermediate permutation instruction;

if said intermediate arrangement is a single monotonically increasing sequence said intermediate arrangement is an initial arrangement, wherein said determined intermediate permutation instructions form a permutation instruction sequence; and

if said intermediate arrangement is not a single monotonically increasing sequence determining a second arrangement for said intermediate arrangement and using said second arrangement as said means for determining a permutation instruction.

33. The system of claim 32 wherein at most lg$n$ said permutation instructions are included in said permutation instruction sequence, wherein n is the number of subwords in said sequence of bits, each said subword comprising one or more bits.

34. The system of claim 32 wherein said programmable processor is a microprocessor, digital signal processor, media processor, multimedia processor, cryptographic processor, network processor, or programmable System-on-a-Chip(SOC).

35. A system of performing an arbitrary permutation of a source sequence of bits in a programmable processor said source sequence of bits is packed into a plurality of source registers comprising the steps of:

means for dividing bits of a first of said source registers to be placed in a first destination register into a first group and bits of said first of said source registers to be placed in a second destination register into a second group with a first GRP permutation instruction;

means for dividing bits of a second of said source registers to be placed in said first destination register into a second group and bits of said second of said source registers to be placed in a second destination register into a first group with a second GRP permutation instruction;

means for placing bits of said first group of said first of said source registers and said bits of said second group of said second of said source registers into said first destination register;

means for placing bits of said second group of said first of said source registers and said first group of said second of said registers into said second destination register;

means for defining a sequence of bits of said first destination register as a first source sequence of bits and a sequence of bits of said second destination register as a second source sequence of bits;

means for defining an intermediate sequence of bits that each of said first source sequence of bits and said second source sequence of bits is transformed into;

means for determining a GRP permutation instruction for transforming said first source sequence of bits and said second source sequence of bits into one or more respective said intermediate sequence of bits until a respective desired sequence of bits is obtained for said first source sequence of bits and said second source sequence of bits,

wherein the determined permutation instructions form a permutation instruction sequence.

36. The system of claim 35 wherein at most 2lg*n* +4 instructions are included in said permutation instruction sequence, wherein *n* is the number of subwords in said sequence of bits, each said subwords comprising one or more bits.

37. The system of claim 35 wherein said programmable processor is a microprocessor, digital signal processor, media processor, multimedia processor, cryptographic processor, network processor, or programmable System-on-a-Chip(SOC).

38. A computer implemented method for performing an arbitrary permutation of a sequence of bits comprising the steps of:

inputting a source sequence of bits into a source register;

defining bit positions in said source sequence of bits to be permuted in said source register for a group of bits in a destination register;

in response to a PPERM instruction inserting bits from said source sequence into said destination register as determined by said bit positions.

39. The method of claim 38 wherein said PPERM instruction comprises a first parameter indicating which *k* bits in said destination register will change, a reference to said source register which contains said source sequence of bits to be permuted, a reference to a configuration register which contains configuration bits for indicating which said bits in said source register are assembled and a reference to said destination register.

40. The method of claim 39 wherein in said destination register said *k* bits specified by said first parameter are updated and all other bits in said destination register are set to zero.

41. The method of claim 39 wherein each of said *k* bits in said final permutation is determined by lg*n* bits to specify which bit in said source register to change.

42. A computer implemented method for performing an arbitrary permutation of a sequence of bits comprising the steps of:

inputting a source sequence of bits into a source register;

defining bit positions in said source sequence of bits to be permuted in said source register for a group of bits in a destination register;

in response to a PPERM3R instruction inserting bits from said source sequence destination register as determined by said bit positions.

43. The method of claim 42 wherein said PPERM3R instruction comprises a first parameter indicating which *k* bits in said destination register will change, a reference to said source register which contains said source sequence of bits to be permuted, a reference to a configuration register which contains configuration bits for indicating which said bits in said source register are assembled and a reference to said destination register.

44. The system of claim 43 wherein in said destination register said *k* bits permuted by said first parameter are updated.

45. The method of claim 43 wherein each of said *k* bits in said final permutation is determined by lg*n* bits to specify which bit in said source register to change.

46. A computer system for performing an arbitrary permutation comprising:

a source register;

a configuration register;

a destination register;

in response to a PPERM instruction placing bits assembled from a sequence of bits from said source register to a position in a sequence of bits in said destination register based on a configuration of bits of said configuration register.

47. The system of claim 46 wherein lg*n*/*k* said PPERM instructions are included in said permutation instruction sequence, wherein *n* is the number of bits in said sequence of bits and *k* is the number of bits that can be changed with one said permutation instruction.

48. A computer readable medium having stored thereon data representing a sequence of permutation instructions, the sequence of permutation instructions which when executed by a processor, cause the processor to permute a source sequence of subwords into one or more intermediate sequences of subwords using a PPERM instruction by placing bits assembled from a sequence of bits from said source register to a position in a sequence of bits in said destination register based on a configuration of bits of said configuration register.

49. The computer readable medium of claim 48 wherein lg*n*/*k* said permutation instructions are included in said permutation instruction sequence, wherein *n* is the number of bits in said sequence of bits and *k* is the number of bits that can be changed with one said permutation instruction.

50. A cryptographic system, having stored thereon data representing a sequence of permutation instructions, the sequence of permutation instructions which when executed by a processor, cause the processor to permute a source sequence of subwords into one or more intermediate sequences of subwords using a PPERM instruction by placing bits assembled from a sequence of bits from said source register to a position in a sequence of bits in said destination register based on a configuration of bits of said configuration register.

51. The cryptographic system of claim 50 wherein lg*n*/*k* said permutation instructions are included in said permutation instruction sequence, wherein *n* is the number of bits in said sequence of bits and *k* is the number of bits that can be changed with one said permutation instruction.

52. A computer system for performing an arbitrary permutation comprising:

a source register;

a configuration register;

a destination register;

in response to a PPERM3R instruction placing bits assembled from a sequence of bits from said source register to a position in a sequence of bits in said destination register based on a configuration of bits of said configuration register.

53. The system of claim 52 wherein lg*n*/*k* said PPERM3R instructions are included in said permutation instruction sequence, wherein *n* is the number of bits in said sequence of bits and *k* is the number of bits that can be changed with one said permutation instruction.

54. A computer readable medium having stored thereon data representing a sequence of permutation instructions, the sequence of permutation instructions which when executed by a processor, cause the processor to permute a source sequence of subwords into one or more intermediate sequences of subwords using a PPERM3R instruction by placing bits assembled from a sequence of bits from said source register to a position in a sequence of bits in said destination register based on a configuration of bits of said configuration register.

55. The computer readable medium of claim 54 wherein lg*n*/*k* said permutation instructions are included in said permutation instruction sequence, wherein *n* is the number of bits in said sequence of bits and *k* is the number of bits that can be changed with one said permutation instruction.

56. A cryptographic system, having stored thereon data representing a sequence of permutation instructions, the sequence of permutation instructions which when executed by a processor, cause the processor to permute a source sequence of subwords into one or more intermediate sequences of subwords using a PPERM3R instruction by placing bits assembled from a sequence of bits from said source register to a position in a sequence of bits in said destination register based on a configuration of bits of said configuration register.

57. The cryptographic system of claim 56 wherein lgn/k said permutation instructions are included in said permutation instruction sequence, wherein n is the number of bits in said sequence of bits and k is the number of bits that can be changed with one said permutation instruction.

58. A computer system for performing an arbitrary permutation comprising:

a source register;

a configuration register;

a destination register;

in response to a GRP instruction dividing an arrangement of a source sequence of bits into a first group and a second group and combining said first group and said second group based on control bits of a configuration register.

59. The system of claim 58 wherein at most lg$n$ said GRP instructions are included in said permutation instruction sequence, wherein n is the number of subwords in said sequence of bits, each said subword comprising one or more bits.

60. A computer readable medium having stored thereon data representing a sequence of permutation instructions, the sequence of permutation instructions which when executed by a processor, cause the processor to permute a source sequence of subwords into one or more intermediate sequences of subwords using a GRP instruction by dividing an arrangement of a source sequence of bits into a first group and a second group and combining said first group and said second group based on control bits of a configuration register.

61. The computer readable medium of claim 60 wherein at most lg$n$ said permutation instructions are included in said permutation instruction sequence, wherein n is the number of subwords in said sequence of bits, each said subword comprising one or more bits.

62. A cryptographic system, having stored thereon data representing a sequence of permutation instructions, the sequence of permutation instructions which when executed by a processor, cause the processor to permute a source sequence of subwords into one or more intermediate sequences of subwords using a GRP instruction by dividing an arrangement of a source sequence of bits into a first group and a second group and combining said first group and said second group based on control bits of a configuration register.

63. The cryptographic system of claim 62 wherein at most lg*n* said permutation instructions are included in said permutation instruction sequence, wherein *n* is the number of subwords in said sequence of bits, each said subword comprising one or more bits.

64. A circuit for implementing a permutation instruction comprising:

a first matrix of a plurality of operation units, each of said operation units comprising a first input connected to a second input, a first output and a second output and a control input, said control input controls a connection between said first input and said second input to said first output or said second output for each of said basic units; and

a second matrix comprising an inversion of said first matrix;

said first matrix being selectively connected to said second matrix.

65. The circuit of claim 64 wherein said permutation instruction is for an *n* bit operation and control bits are encoded by one-hot code for combining results of two *n* bit operations into a 2*n* bit operation.

66. A method of performing an arbitrary permutation in a programmable processor comprising the steps of:

a. defining bit positions in a source sequence of bits to be permuted in a source register for a group of bits in a destination register, said source sequence being stored in a plurality of source registers;

b. determining a permutation instruction with said bit positions to assemble bits from said source sequence of bits;

c. performing said permutation instruction for inserting said assembled bits into a destination register as determined by said bit positions; and

d. repeating steps a. through c. for groups of bits in said destination register,

wherein after a final permutation instruction a desired permutation of said source register is determined and said determined permutation instructions form a permutation instruction sequence.

67. The method of claim 66 wherein step d repeats steps a. through c. for all non-overlapping said groups of bits in said destination register.

68. The method of claim 66 wherein said permutation instruction is a PPERM instruction which comprises a first parameter indicating which *k* bits in said destination register will change, a second parameter for indicating an identification of which of said source registers a subset of said source sequence of bits is stored, a reference to said source register which contains said source sequence of bits to be permuted, a reference to one or more configuration registers which contain configuration bits for indicating which said bits in said source register are assembled and one or more index bits for selecting each said bit in said source register and one or more bits for indicating an identification for each said index bits and a reference to said destination register.

69. The method of claim 68 wherein said destination registers said k bits specified by said first parameter are updated if said second parameter and said identification for each of said configuration bits is the same.

70. The method of claim -66 wherein said permutation instruction is a PPERM3R instruction which comprises a first parameter indicating which *k* bits in said destination register will change, a reference to said source register which contains said source sequence of bits to be permuted, a reference to one or more configuration registers which contain configuration bits for indicating which said bits in said source register are assembled and one or more index bits for selecting each said bit in said source register and one or more bits for indicating an identification for each said index bits and a reference to said destination register.

71. The method of claim 70 wherein said destination registers said k bits specified by said first parameter are updated if said second parameter and said identification for each of said configuration bits is the same.

72. The method of claim 66 wherein said permutation instruction is a PPERM instruction which comprises a first parameter indicating which *k* bits in said destination register will change, a reference to said source register which contains said source sequence of bits to be permuted, a reference to one or more configuration registers which contain configuration bits for indicating which said bits in said source register are assembled and said configuration bits including one or more control bits for selecting each said bit in said source register.

73. The method of claim 72 wherein if said control bit is a 0 placing said bit in said source register into said destination register in accordance with said configuration bits and if said control bit is a 1 said bit in said source register is set to 0.

74. The method of claim 66 wherein said permutation instruction is a PPERM3R instruction which comprises a first parameter indicating which *k* bits in said destination register will change, a reference to one or more configuration registers which contain configuration bits for indicating which said bits in said source register are assembled and said configuration bits including one or more control bits for selecting each said bit in said source register.

75. The method of claim 74 wherein if said control bit is a 0 selecting said bit in said source register and placing said selected bit in said source register into said destination register in accordance with said configuration bits and if said control bit is a 1 said bit in said source register moved into said destination register unchanged.

76. A method of performing an arbitrary permutation in a programmable processor comprising the steps of:

a. defining bit positions in a source sequence of bits to be permuted in a source register for a group of bits in a destination register;

b. determining a permutation instruction with said bit positions to assemble bits from said source sequence of bits;

c. performing said permutation instruction for inserting said assembled bits into a destination register as determined by said bit positions; and

d. repeating steps a. through c. for groups of bits in said destination register,

wherein after a final permutation instruction a desired permutation of said source register is determined and said determined permutation instructions form a permutation instruction sequence and said permutation has bit repetitions.

77. A method of performing an arbitrary permutation in a programmable processor comprising the steps of:

a. defining bit positions in a source sequence of bits to be permuted in a source register for a group of bits in a destination register;

b. determining a permutation instruction with said bit positions to assemble bits from said source sequence of bits;

c. performing said permutation instruction for inserting said assembled bits into a destination register as determined by said bit positions;

d. repeating steps a. through c. for groups of bits in said destination register, after a final permutation instruction a desired permutation of said source register is determined and said determined permutation instructions form a permutation instruction sequence: and

e. executing at least one other instruction interspersed with said determined permutation instructions during execution of said permutation instruction sequence.